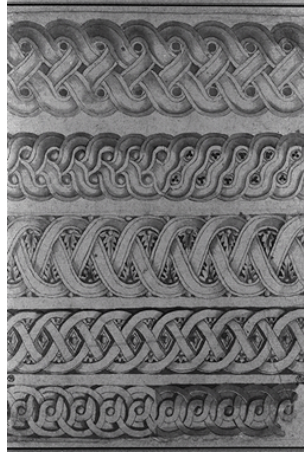


# Assignment 2

COMP9021, Trimester 2, 2020



## 1. GENERAL MATTERS

1.1. **Aim.** The main purpose of the assignment is to:

- develop your problem solving skills;
- write a medium sized Python program;
- design and implement an interface based on the desired behaviour of an application program;
- work with classes.

1.2. **Submission.** Your programs will be stored in a file named `frieze.py`. After you have developed and tested your program, upload it using Ed (unless you worked directly in Ed). Assignments can be submitted more than once; the last version is marked. Your assignment is due by August 9, 8:00pm.

1.3. **Assessment.** The assignment is worth 13 marks. It is going to be tested against a number of inputs. For each test, the automarking script will let your program run for 30 seconds.

Late assignments will be penalised: the mark for a late submission will be the minimum of the awarded mark and 13 minus the number of full and partial days that have elapsed from the due date.

The outputs of your programs should be **exactly** as indicated.

1.4. **Reminder on plagiarism policy.** You are permitted, indeed encouraged, to discuss ways to solve the assignment with other people. Such discussions must be in terms of algorithms, not code. But you must implement the solution on your own. Submissions are routinely scanned for similarities that occur when students copy and modify other people's work, or work very closely together on a single implementation. Severe penalties apply.



3.2. **Second example.** The file `frieze_2.txt` has the following contents.

```

4 4 4 4 4 4 4 4 4 12 4 4 4 4 4 4 4 4 4 12 4 4 4 4 4 4 4 4 4 12 4 4 4 4 4 4 4 4 4 12 4 0
0 0 0 4 8 0 0 0 3 1 1 1 0 0 0 4 8 0 0 0 3 1 1 1 0 0 0 4 8 0 0 0 3 1 1 1 0 0 0 4 8 0 0 0 3 1 1 1 0
0 0 2 0 0 0 0 0 0 1 1 0 0 0 2 0 0 0 0 0 0 1 1 0 0 0 2 0 0 0 0 0 0 1 1 0 0 0 2 0 0 0 0 0 0 1 1 0 0
1 0 9 0 0 1 0 1 0 1 1 0 1 0 9 0 0 1 0 1 0 1 1 0 1 0 9 0 0 1 0 1 0 1 1 0 1 0 9 0 0 1 0 1 0 1 1 0 1
4 0 0 4 2 0 4 4 8 1 1 4 4 0 0 4 2 0 4 4 8 1 1 4 4 0 0 4 2 0 4 4 8 1 1 4 4 0 0 4 2 0 4 4 8 1 1 4 0
8 1 0 0 0 0 1 0 0 1 3 0 8 1 0 0 0 0 1 0 0 1 3 0 8 1 0 0 0 0 1 0 0 1 3 0 8 1 0 0 0 0 1 0 0 1 3 0 0
1 1 0 4 0 0 3 1 0 0 0 0 1 1 0 4 0 0 3 1 0 0 0 0 1 1 0 4 0 0 3 1 0 0 0 0 1 1 0 4 0 0 3 1 0 0 0 0 1
1 0 0 5 1 0 0 1 0 4 0 0 1 0 0 5 1 0 0 1 0 4 0 0 1 0 0 5 1 0 0 1 0 4 0 0 1 0 0 5 1 0 0 1 0 4 0 0 1
1 4 4 0 4 4 8 1 0 1 1 0 1 4 4 0 4 4 8 1 0 1 1 0 1 4 4 0 4 4 8 1 0 1 1 0 1 4 4 0 4 4 8 1 0 1 1 0 1
3 0 8 1 1 0 0 1 0 1 1 0 3 0 8 1 1 0 0 1 0 1 1 0 3 0 8 1 1 0 0 1 0 1 1 0 3 0 8 1 1 0 0 1 0 1 1 0 1
0 0 1 1 3 1 0 0 4 1 5 0 0 0 1 1 3 1 0 0 4 1 5 0 0 0 1 1 3 1 0 0 4 1 5 0 0 0 1 1 3 1 0 0 4 1 5 0 0
0 0 1 0 8 1 0 0 1 0 0 1 0 0 1 0 8 1 0 0 1 0 0 1 0 0 1 0 8 1 0 0 1 0 0 1 0 0 1 0 8 1 0 0 1 0 0 1 0
4 4 7 5 5 5 4 4 5 4 4 5 4 4 7 5 5 5 4 4 5 4 4 5 4 4 7 5 5 5 4 4 5 4 4 5 4 4 7 5 5 5 4 4 5 4 4 5 0

```

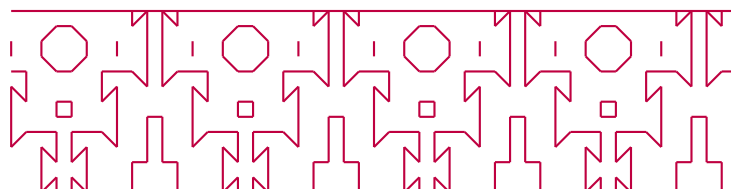
Here is a possible interaction:

```

$ python3
...
>>> from frieze import *
>>> frieze = Frieze('frieze_2.txt')
>>> frieze.analyse()
Pattern is a frieze of period 12 that is invariant under translation
        and vertical reflection only.
>>> frieze.display()

```

The effect of executing `frieze.display()` is to produce a file named `frieze_2.tex` that can be given as argument to `pdflatex` to produce a file named `frieze_2.pdf` that views as follows.







3.5. **Fifth example.** The file `frieze_5.txt` has the following contents.

```

4 4 4 4 4 4 12 4 4 4 4 4 4 12 4 4 4 4 4 4 12 4 4 4 4 4 4 12 4 4 4 4 4 4 12 4 0
0 4 1 0 0 2 0 8 0 4 1 0 0 2 0 8 0 4 1 0 0 2 0 8 0 4 1 0 0 2 0 8 0 4 1 0 0 2 0 8 0
10 0 0 0 2 0 0 0 10 0 0 0 2 0 0 0 10 0 0 0 2 0 0 0 10 0 0 0 2 0 0 0 10 0 0 0 2 0 0 0
0 8 0 2 0 0 4 2 0 8 0 2 0 0 4 2 0 8 0 2 0 0 4 2 0 8 0 2 0 0 4 2 0 8 0 2 0 0 4 2 0
4 4 6 4 4 4 5 4 4 4 6 4 4 4 5 4 4 4 6 4 4 4 5 4 4 4 6 4 4 4 5 4 4 4 6 4 4 4 5 4 0

```

Here is a possible interaction:

```

$ python3
...
>>> from frieze import *
>>> frieze = Frieze('frieze_5.txt')
>>> frieze.analyse()
Pattern is a frieze of period 8 that is invariant under translation
      and rotation only.
>>> frieze.display()

```

The effect of executing `frieze.display()` is to produce a file named `frieze_5.tex` that can be given as argument to `pdflatex` to produce a file named `frieze_5.pdf` that views as follows.



3.6. **Sixth example.** The file `frieze_6.txt` has the following contents.

```

4 4 4 4 4 4 4 4 4 4      4 4 4 4 4 4 4 4 4 4      4 4 4 4 4 4 4 4 4 4      4 4 4 4 4 4 4 4 4 4      0
4 0 4 4 4 4 0 4 4 4 0    4 4 4 0 4 4 4 0 4 4      4 0 4 4 4 0 4 4 4 0      4 4 4 0 4 4 4 0 4 4      0
0 1 1 4 0 1 1 4 0 1      1 4 0 1 1 4 0 1 1 4      0 1 1 4 0 1 1 4 0 1      1 4 0 1 1 4 0 1 1 4      0
1 5 1 1 1 5 1 1 1 5      1 1 1 5 1 1 1 5 1 1      1 5 1 1 1 5 1 1 1 5      1 1 1 5 1 1 1 5 1 1      1
5 4 4 1 5 4 4 1 5 4      4 1 5 4 4 1 5 4 4 1      5 4 4 1 5 4 4 1 5 4      4 1 5 4 4 1 5 4 4 1      1

4 4 4 4 4 4 4 4 4 4      4 4 4 4 4 4 4 4 4 4      4 4 4 4 4 4 4 4 4 4      4 4 4 4 4 4 4 4 4 4      0

```

Here is a possible interaction:

```

$ python3
...
>>> from frieze import *
>>> frieze = Frieze('frieze_6.txt')
>>> frieze.analyse()
Pattern is a frieze of period 4 that is invariant under translation,
        glided horizontal and vertical reflections, and rotation only.
>>> frieze.display()

```

The effect of executing `frieze.display()` is to produce a file named `frieze_6.tex` that can be given as argument to `pdflatex` to produce a file named `frieze_6.pdf` that views as follows.



3.7. **Seventh example.** The file `frieze_7.txt` has the following contents.

```

4 4 12 4 4 4 12 4 4 4 12 4 4 4 12 4 4 4 12 4 0
    0 2 0 8 0 2 0 8 0 2 0 8 0 2 0 8 0 2 0 8 0
10 0 8 0 10 0 8 0 10 0 8 0 10 0 8 0 10 0 8 0 0
    0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0 2 0
    0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0
    0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0 9 0
10 0 2 0 10 0 2 0 10 0 2 0 10 0 2 0 10 0 2 0 0
    0 8 0 2 0 8 0 2 0 8 0 2 0 8 0 2 0 8 0 2 0
    4 4 6 4 4 4 6 4 4 4 6 4 4 4 6 4 4 4 6 4 0

```

Here is a possible interaction:

```

$ python3
...
>>> from frieze import *
>>> frieze = Frieze('frieze_7.txt')
>>> frieze.analyse()
Pattern is a frieze of period 4 that is invariant under translation,
        horizontal and vertical reflections, and rotation only.
>>> frieze.display()

```

The effect of executing `frieze.display()` is to produce a file named `frieze_7.tex` that can be given as argument to `pdflatex` to produce a file named `frieze_7.pdf` that views as follows.





#### 4. DETAILED DESCRIPTION

**4.1. Input.** The input is expected to consist of  $height + 1$  lines of  $length + 1$  numbers in  $\{0, \dots, 15\}$ , where  $length$  is at least equal to 4 and at most equal to 50 and  $height$  is at least equal to 2 and at most equal to 16, with possibly lines consisting of spaces only that will be ignored and with possibly spaces anywhere on the lines with numbers. The  $x^{th}$  number  $n$  of the  $y^{th}$  line, with  $0 \leq x \leq length$  and  $0 \leq y \leq height$ ,

- is to be associated with a point situated  $x \star 0.2$  cm to the right and  $y \star 0.2$  cm below an origin,
- is to be connected to the point 0.2 cm above if the rightmost digit of  $n$  is 1,
- is to be connected to the point 0.2 cm above and 0.2 cm to the right if the second rightmost digit of  $n$  is 1,
- is to be connected to the point 0.2 cm to the right if the third rightmost digit of  $n$  is 1, and
- is to be connected to the point 0.2 cm to the right and 0.2 cm below if the fourth rightmost digit of  $n$  is 1.

To qualify as a frieze, the input is further constrained to fit in a rectangle of length  $length \star 0.2$  cm and of height  $height \star 0.2$  cm, with horizontal lines of length  $length$  at the top and at the bottom, identical vertical borders at both ends, no crossing segments connecting pairs of neighbours inside the rectangle, and a pattern of integral period at least equal to 2 that is fully repeated at least twice in the horizontal dimension.

**4.2. Output.** Consider executing from the Python prompt the statement `from frieze import *` followed by the statement `frieze = Frieze(some_filename)`. In case `some_filename` does not exist in the working directory, then Python will raise a `FileNotFoundError` exception, that does not need to be caught. Assume that `some_filename` does exist (in the working directory). If the input is incorrect in that it does not contain only numbers in  $\{0, \dots, 15\}$  besides spaces, or in that it contains either too few or too many lines of numbers, or in that some line of numbers contains too many or too few numbers, or in that two of its lines of numbers do not contain the same number of numbers, then the effect of executing `frieze = Frieze(some_filename)` should be to generate a `FriezeError` exception that reads

```
Traceback (most recent call last):
```

```
...
```

```
frieze.FriezeError: Incorrect input.
```

If the previous conditions hold but the further conditions spelled out above for the input to qualify as a frieze do not hold, then the effect of executing `frieze = Frieze(some_filename)` should be to generate a `FriezeError` exception that reads

```
Traceback (most recent call last):
```

```
...
```

```
frieze.FriezeError: Input does not represent a frieze.
```

If the input is correct and represents a frieze, then executing `frieze = Frieze(some_filename)` followed by `frieze.analyse()` should have the effect of outputting one or two lines that read

```
Pattern is a frieze of period N that is invariant under translation only.
```

```
or
```

```
Pattern is a frieze of period N that is invariant under translation
and vertical reflection only.
```

```
or
```

```
Pattern is a frieze of period N that is invariant under translation
and horizontal reflection only.
```

```
or
```

```
Pattern is a frieze of period N that is invariant under translation
and glided horizontal reflection only.
```

```
or
```

```
Pattern is a frieze of period N that is invariant under translation
and rotation only.
```

```
or
```

Pattern is a frieze of period  $N$  that is invariant under translation,  
glided horizontal and vertical reflections, and rotation only.

or

Pattern is a frieze of period  $N$  that is invariant under translation,  
horizontal and vertical reflections, and rotation only.

with  $N$  an appropriate integer at least equal to 2.

These 7 possible outputs are based on a mathematical result on the classification of friezes that lists all possible complete lists of symmetries that leave a frieze invariant under an isometry (that is, a transformation that does not alter the distance between any two points). These possible lists involve 5 symmetries.

- Translation by *period*; of course, any frieze is invariant under this symmetry.
- Vertical reflection about some vertical line; that line does not necessarily delimit the pattern nor does it necessarily go through its middle (these conditions are actually equivalent).
- Horizontal reflection about the line that goes through the middle of the frieze.
- Glided horizontal reflection, that is, horizontal reflection about the line that goes through the middle of the frieze and translation by half the period of the resulting lower half of the frieze.
- Rotation around some point situated on the horizontal line that goes through the middle of the frieze; this is equivalent to horizontal reflection combined with vertical reflection.

Pay attention to the expected format, including spaces.

If the input is correct and represents a frieze, then executing `frieze = Frieze(some_filename)` followed by `frieze.display()` should have the effect of producing a file named `some_filename.tex` that can be given as argument to `pdflatex` to generate a file named `some_filename.pdf`. The provided examples will show you what `some_filename.tex` should contain. Segments are drawn in purple with a single *draw* command for each longest segment,

- starting with the vertical segments, from the topmost leftmost one to the bottommost rightmost one with the leftmost ones first,
- followed by the segments that go from north west to south east, from the topmost leftmost one to the bottommost rightmost one with the topmost ones first,
- followed by the segments that go from west to east, from the topmost leftmost one to the bottommost rightmost one with the topmost ones first,
- followed by the segments that go from the south west to the north east, from the topmost leftmost one to the bottommost rightmost one with the topmost ones first.

Pay attention to the expected format, including spaces and blank lines. Lines that start with `%` are comments; there are 4 such lines, that have to be present even when there is no item to be displayed of the kind described by the comment. The output of your program redirected to a file will be compared with the expected output saved in a file (of a different name of course) using the `diff` command. For your program to pass the associated test, `diff` should silently exit, which requires that the contents of both files be absolutely identical, character for character, including spaces and blank lines. When testing locally, check your program on the provided examples using the associated `.tex` files, renaming them as they have the names of the files expected to be generated by your program.